Facial Geometry Parameterisation Based on Partial Differential Equations

Yun Sheng[a,1], Phil Willis[a], Gabriela Gonzalez Castro[b] and Hassan Ugail[b]

[a] *Department of Computer Science, University of Bath, UK*

[b] *School of Informatics, University of Bradford, UK*

Abstract: Geometric modelling using Partial Differential Equations (PDEs) has been gradually recognised due to its smooth instinct, as well as the ability to generate a variety of geometric shapes by intuitively manipulating a relatively small set of PDE boundary curves. In this paper we explore and demonstrate the feasibility of the PDE method in facial geometry parameterisation. The geometry of a generic face is approximated by evaluating spectral solutions to a group of fourth order elliptic PDEs. Our PDEs-based parameterisation scheme can produce and animate a high-resolution 3D face with a relatively small number of parameters. By taking advantage of parametric representation, the PDE method can use one fixed animation scheme to manipulate the facial geometry in varying Levels of Detail (LODs), without any further process.

Keywords: Computational Geometry, Partial Differential Equation, Face Modeling, MPEG-4.

Classification Codes (a-ccording to the standard AMS codes):

35 Partial Differential Equations

35J38 Boundary value problems for high-order elliptic systems

35C05 Solutions in closed form

35Q68 PDEs in connection with computer science

[1] Present affiliation of corresponding author: School of Computer Engineering, Nanyang Technological University, Singapore. Email: shengyun@ntu.edu.sg
Tel: +65 82128971 Fax: +65 67928123

1. Introduction

The subject of Partial Differential Equations (PDEs) emerged in the 18th century as ordinary differential equations failed to describe some physical phenomena. Since then, many physical phenomena and paramount discoveries have been branded with the PDEs. Geometric modelling using PDEs has been widely studied in computer graphics since Bloor *et al*'s PDE method was first introduced in blend surface generation two decades ago [1]. PDE methods adopt a boundary value approach whereby 3D geometric models can be reconstructed by solving PDEs either analytically or numerically with relevant boundary conditions. Advantages of the PDE method have been gradually recognised by researchers. A principal advantage comes from the ability that differential operators of the PDEs can ensure a generation of smooth surfaces, where the smoothness is strictly governed by the order of the PDEs used. A second advantage of using the PDE methods is that a PDE surface can be generated by intuitively manipulating a relatively small set of boundary curves. Moreover, the behaviour of PDE surfaces has been proven compatible with those underlying tensor-product surfaces, such as Bezier surface [2], B-splines [3], *etc*. Taken together, these advantages have contributed to a widespread adoption of the PDE methods in a range of disciplines, such as free-form surface design [4], solid modelling [5], computer aided manufacturing [6,7], shape morphing [8], web visualisation [9], mesh reconstruction [10], etc. In this paper, we address a prolonged topic, facial geometry parameterisation, by exploring and examining the feasibility of one PDE method in face modelling and facial animation.

Facial geometry parameterisation has been playing a crucial role in many fields, such as in the film industry and the games industry where 3D graphical models are used to enhance vividness and performance, in telecommunication where a 3D face is employed to improve coding efficiency for video conferencing [11], and even in medical operation where the 3D face model can be utilised to simulate plastic surgery. Efficient modelling of the human face and precise simulation of its expression rely on a sophisticated parameterisation procedure. To model the face realistically, most the existing parameterisation schemes utilise a fixed high-resolution face model, which is operationally clumsy and computationally complex in storage, loading and animation. We propose to parameterise the facial geometry based on the use of the PDEs, where the geometry of a generic face is approximated by evaluating spectral solutions to a group of fourth order elliptic PDEs. The developed PDEs-based geometry parameterisation scheme can produce and animate a high-resolution 3D face with a relatively small number of parameters only. For instance, smooth animation can be produced by manipulating a small group of Facial Animation Parameters (FAPs) [12], which are associated with a fixed number of

feature points on the PDE boundary curves. By taking advantage of parametric representation the PDE method can use one fixed animation scheme, or one Facial Animation Table (FAT) [13] to manipulate the facial geometry in varying Levels of Detail (LODs), without any extra process. Moreover, its smooth instinct ensures that the PDE method can reconstruct smooth face surfaces with desired continuities.

The paper unfolds as follows. Section 2 briefly surveys the research literature. Section 3 theoretically introduces the PDE method used in this paper with a spectral approximation to its solution. Section 4 is dedicated to the PDE face, a 3D face model created by the PDE method and the PDEs-driven animation is introduced in Section 5. Section 6 demonstrates the feasibility of the developed geometry parameterisation scheme by carrying out some expression synthesis and transfer tests. In Section 7 we discuss several technical issues existing in the developed parameterisation scheme.

2. Related work

2.1. Facial geometry parameterisation

The earliest attempt in facial geometry parameterisation dates back to the early 1970's when Parke's pioneer work, a parameterised model, was initially used to replace a limited conventional technique, called key frame animation, in facial animation [14]. The creation of his model involved two basic initiatives: (1) To determine an appropriate set of parameters for facial expressions; (2) to develop image synthesis models based on these parameters. Since then, considerable effort has been made to parameterise the facial geometry more realistically with a lower computational complexity. For example, the CANDIDE face model series [15] has been widely exploited because of their simplicity and public availability [16]. Moreover, the Instituto Superior Tecnico also defined a simply textured 3D head model [17], developed from Parke's well-known parameterised model, which has been embedded in many MPEG-4 players by researchers.

Another class of face models, called physical face models, have been developed in accordance with the physical and anatomical knowledge of the human face. Well-known examples embrace the Facial Animation, Construction and Editing System (FACES), a software simulator of the human face [18] and Waters's vector approach to simulate the facial muscles [19]. Later, Terzopoulos et al. proposed a three-layer face model for modeling the detailed anatomical structure and dynamics of the human face [20]. The three layers to simulate the skin tissue correspond to the epidermis, dermal layer and muscle layer. Skin properties and muscle

actions are simulated using elastic springs that connect each layer. Muscle forces propagate through the mesh systems to create animation.

With the advance of the parameterised face models, parameterising facial animation has become feasible and can be implemented by a small set of animation parameters. Before MPEG-4 was standardized, the Facial Action Coding System (FACS) [21] was extensively used. This system was initially used in psychological study rather than in facial modelling. The FACS defines basic facial muscle movement in terms of 44 Action Units (AUs), based on an analysis of the facial anatomy. Any facial expression can be formed by a combination of these AUs. One deficiency of the AUs is that the FACS failed to define how and where to move 3D vertices of the face model. Rydfalk parameterised these AUs with his CANDIDE model [22], where the motion of each AU was described as an interaction of a set of vertices in the model. The MPEG-4 standard defines 68 Facial Animation Parameters (FAPs), containing 2 high-level parameters, viseme and expression, 66 low-level parameters associated with minimal facial actions. The viseme represents the physical configuration of the mouth, tongue and jaw that is visually correlated with the speech sound corresponding to the phoneme, while the expression parameter allows a definition of high-level facial expressions, enabling efficient facial animation. On the other hand, each FAP of the 66 low-level parameters directly corresponds to the local movement of an individual FDP (Face Definition Parameter) feature point [13]. In Parke's book [23], an approach that utilises a set of animation parameters, such as the AUs or MPEG-4 FAPs, to parameterise facial animations is referred to as the direct parameterisation method. The direct parameterisation method is among those most commonly used animation approaches because it provides a parametric description of the face geometry, with a high degree of adaptability to applications.

For the sake of realism, researchers prefer to use high-resolution face models nowadays, typically those produced by 3D scanning with more than thousands of vertices [24,25,26]. These models require a heavy load in manipulation and animation. To animate such face models with a direct parameterisation scheme, one is required to identify a set of control points out of the original mesh to represent the most salient facial features. Facial animation relies on parameterising this relatively small number of control points. These control points are then used to move their surrounding vertices via surface interpolation algorithms, such as the radial basis function, Dirichlet free-form deformation [25] or a resampling method [26]. This raises a new problem of how to make the motion of the control points influence the surrounding vertices smoothly and effectively. Moreover, most high-resolution face models offer only one fixed LOD in representation, the mesh of which may need to be scaled down to a low resolution for a low

storage cost or during a limited bandwidth transmission. There are some mesh simplification algorithms [27] able to reduce the mesh resolutions. These algorithms are based on edge collapsing, vertex removal and mesh merging, which will modify the original connectivity of the meshes so that the direct parameterisation algorithms cannot associate the animation parameters with the same group of feature vertices on the original face model in varying LODs, leading to a computational complexity. More detailed surveys can be found in [28] [29].

2.2. PDE methods

Since Bloor *et al*'s pioneering work first made use of the PDEs in blend surface generation two decades ago [1], the benefits of using the PDEs have been gradually recognised by many researchers, in order to address a variety of problems, such as free-form surface design [4], solid modelling [5], interactive design [30] and so forth. One major use of the Bloor *et al*'s PDE method was to develop computer aided manufacturing tools by which many industrial models, such as the marine propeller [7], the swirl port [31] and aircraft geometry [6], can be easily mimicked. Moreover, it has also been demonstrated that the PDE method can be applied to shape morphing [8], geometric modelling for web visualisation [9], mesh reconstruction [10], *etc*.

You *et al* introduced a fast approximate analytic solution to a fourth order PDE using the Pseudo-Levy series [32]. Normally, the fourth order PDE is believed to be accurate enough to assure a smooth surface within a PDE patch, while a higher order is more time-consuming. In 2004, aiming to increase the continuity between the joint PDE patches of 3D models, You *et al* proposed to extend the PDEs to a sixth order, so as to achieve C^2 curvature continuity between the joint surface patches [33]. This was also proven by Kubiesa *et al* in the same year [34].

All the PDE methods surveyed above are based upon an analytic solution to resolve the PDEs. The analytic solutions suit for those PDEs with close-form boundary conditions. Otherwise, a solution will have to be sought numerically, but this also means an increase in computational load. Du and Qin proposed to use the finite difference method to resolve the PDEs [35]. Because not all the PDEs have an analytic solution, the numerical solution may improve the generality of the PDE method, but will bring a slow processing speed. In [5], Du and Qin used a free form deformation technique for solid design. One major contribution of this work was that they coined the notion of the 4D PDE, which was made up not only of a 3D parametric representation for solid geometry, but also of an additional 1D implicit representation describing solid properties, such as the material intensity, *etc*.

3. A spectral approximation to PDE surfaces

In this paper we adopt a PDE method by analytically solving fourth order PDEs, the solutions of which determine the underlying 3D face reconstruction. We use a spectral method to approximate the PDE solution for an efficient evaluation [36].

3.1. The PDE method

The PDE method adopted in this paper produces a parametric surface $S(u, v)$, where $S(u, v) = [x(u, v), y(u, v), z(u, v)]$ in Cartesian coordinates, defined as the solution to a biharmonic-like fourth order elliptic PDE:

$$\left(\frac{\partial^2}{\partial u^2} + a^2 \frac{\partial^2}{\partial v^2}\right)^2 S(u,v) = 0 \tag{1}$$

where u and v are the parametric surface independent variables. The partial differential operators in Equation 1 represent a smoothing process in which the value of the function at any point on the surface is, in a certain sense, a weighted average of the surrounding values. In this way a surface is obtained as a smooth transition between the boundary conditions. The parameter a ($a \geq 1$), called the smoothing parameter, controls the relative rates of smoothing between the u and v parameter directions. Viewed differently, the parameter a determines the rate at which the solution "forgets" about the boundaries as one moves away from a boundary into the surface interior. Thus, by adjusting the value of a, one is able to control how much the boundary conditions influence the interior of the surface.

The reason of using the fourth order PDE is that a lower order PDE provides no freedom to specify both S and its derivatives on the boundary, which is necessary if there is to be tangent or C^1 continuity between the blend and adjacent surfaces; whereas the higher orders are more time-consuming.

3.2. A spectral approximation to PDE solution

Equation 1 can be resolved either analytically or fully numerically. Analytic solutions suit for those PDEs with close-form boundary conditions. Otherwise, a solution will have to be sought numerically, but this also means an increase in computational load. In our case, we choose an analytic method to evaluate the PDEs since it is likely to represent PDE boundary conditions of the human head in close-form, and computationally this makes calculation and recalculation in response to user manipulation and interaction, very efficient. The analytic method adopted here uses a spectral approximation to the PDE solution which satisfies the

associated boundary conditions, and expresses the solution in terms of a finite sum of analytic functions with a remainder term [36].

Briefly, assuming that the effective region in uv space is restricted to $0 \le u \le 1$ and $0 \le v \le 2\pi$, the PDE algorithm exploits an analytical solution, which is of the form

$$S(u,v) = A_0(u) + \sum_{n=1}^{\infty} \left[A_n(u)\cos(nv) + B_n(u)\sin(nv) \right] \tag{2}$$

where

$$A_0(u) = \alpha_{00} + \alpha_{01}u + \alpha_{02}u^2 + \alpha_{03}u^3 \tag{3}$$

$$A_n(u) = \alpha_{n1}e^{anu} + \alpha_{n2}ue^{anu} + \alpha_{n3}e^{-anu} + \alpha_{n4}ue^{-anu} \tag{4}$$

$$B_n(u) = \beta_{n1}e^{anu} + \beta_{n2}ue^{anu} + \beta_{n3}e^{-anu} + \beta_{n4}ue^{-anu} \tag{5}$$

where α_{00}, α_{01}, …, α_{n3}, α_{n4} and β_{11}, β_{12},…, β_{n3}, β_{n4} are vector-valued coefficients. In Equation 2 the PDE solution is expressed in terms of an infinite sum of analytic function modes which individually satisfies the PDE. The amplitude of the mode decays as its frequency increases. Thus, it is not difficult to prove that the first N low frequency modes containing the most essential geometric information are the major contributors to the resulting surface while the remaining high frequency modes are trivial enough to be neglected as long as the value of N is properly selected. In a typical problem, 5 modes are more than adequate. Therefore, Equation 2 can be rewritten as

$$S(u,v) = A_0(u) + \sum_{n=1}^{N} \left[A_n(u)\cos(nv) + B_n(u)\sin(nv) \right] + \sum_{n=N+1}^{\infty} \left[A_n(u)\cos(nv) + B_n(u)\sin(nv) \right] \tag{6}$$

For a given set of boundary conditions, in order to determine the various constants in the solution, it is necessary to Fourier analyse the boundary conditions and identify the various Fourier coefficients with the values of A_0, A_n and B_n. The basic idea is to approximate the solution to Equation 6 which is a periodic Fourier series expansion for $v \in (-\infty, \infty)$, by the sum of a finite number of Fourier modes and a remainder term:

$$S(u,v) = A_0(u) + \sum_{n=1}^{N} \left[A_n(u)\cos(nv) + B_n(u)\sin(nv) \right] + R(u,v) \tag{7}$$

where the remainder term R is defined as:

$$R(u,v) = r_1(v)e^{wu} + r_2(v)e^{-wu} + r_3(v)ue^{wu} + r_4(v)ue^{-wu} \tag{8}$$

with $w = a\,(N + 1)$. Let C_1, C_2, C_3 and C_4 indicate the four boundary conditions imposed on the PDE. Fourier analysis of the boundary conditions permits the following expression:

$$C_i(v) = a_{0i} + \sum_{n=1}^{\infty} \left[a_{ni}\cos(nv) + b_{ni}\sin(nv) \right] \tag{9}$$

where i=1, 2, 3 and 4. By comparison of Equations 2 and 9, we can easily identify $a_{01} = A_0(0)$, $a_{n1} = A_n(0)$, $b_{n1} = B_n(0)$ and so forth. Thus, combining Equation 7 and 9, we can write four equations for calculation of the remainder terms of the four boundary conditions:

$$R_{c1} = C_1 - F_{c1};\ R_{c2} = C_2 - F_{c2};$$
$$R_{c3} = C_3 - F_{c3};\ R_{c4} = C_4 - F_{c4}; \tag{10}$$

where

$$F(u,v) = A_0(u) + \sum_{n=1}^{N}\left[A_n(u)\cos(nv) + B_n(u)\sin(nv)\right] \tag{11}$$

Consequently functions $r_1(v)$, $r_2(v)$, $r_3(v)$ and $r_4(v)$ in Equation 8 can be solved using Equation 10.

3.3. Boundary conditions

Boundary conditions imposed around the edges of a surface patch control the internal shape of the surface. The question of how the boundary conditions should be chosen is paramount in the PDE method. In generating a free form surface, one has considerable freedom to choose boundary conditions in order to achieve a desired shape. In the PDE method, boundary conditions are usually associated with the boundary curves, a set of isoparametric lines in 3D space. To resolve the fourth order PDE, there need to be four boundary conditions so as to obtain a unique solution. In [36], the four boundary conditions imposed on the PDEs consist of $S(0, v)$ and $S(1, v)$, the positional boundary curves at $u = 0$ and 1, and $S_u(0, v)$, $S_u(1, v)$, the two derivative conditions along the u direction. The derivative conditions play an important role in determining the overall shape of the surface, and they control the direction and intensity in which the surface leaves the boundary curves.

With derivative conditions, the resulting surface will not pass through all the four boundary curves. In order to make the PDE surface pass through all the boundary curves, the four boundary curves can be directly employed as the boundary conditions for the PDEs. This provides users with more precise control of the resulting surface, facilitating the interactive manipulation.

4. PDE face

A 3D face model reconstructed by the PDE method is referred to here as a PDE face. As we have known, creating a 3D surface patch using a fourth-order PDE requires four boundary conditions. When modelling a more complex object like the face, a set of PDE patches with four boundary conditions for each patch are needed so as to represent the facial geometry precisely.

As we discussed in the previous section, the direct use of four boundary curves as four boundary conditions to the PDE solution can make sure that the resulting PDE surface pass through all the boundary curves, and can provide users with more precise control of the resulting surface, facilitating the interactive manipulation. As our primary goal is to parameterise the face precisely, we compromise the derivative conditions at this stage such that the surface of the reconstructed PDE face will pass through all the boundary curves. Not only does this give the generated face a high fidelity, but also this can provide accurate control during facial animation because some of the facial curves will be identified as facial features.

4.1. PDE face design

As the PDE method is a boundary value approach, the selection of the PDE face boundary curves is crucial that must best describe the face geometry. Unlike the PDE methods in free-form surface design where its boundary curves may be drawn with a high degree of freedom, the PDE face boundary curves must abide by some constraints. For example, the face boundary curves have to be or close to be symmetric with respect to the nose bridge. In order to obtain a generic group of facial boundary curves, we carry out the following operations: (a) Use a 3D laser scanner to capture a series of human faces in a consistent neutral pose, *i.e*. with the eyes open and the mouth closed, resulting in a set of 3D data for the faces. (b) Extract consistently the same number of boundary curve points from each of the 3D data. (c) Calculate an average position of the extracted boundary curves for the generic facial boundary curves.

The PDE face is created by calculating nine different facial surface patches, each of which is generated by four consecutive boundary curves. All the patches share one common boundary curves with each other, apart from Patch 6 and 7 (the numbering shown in Figure 1). Totally, there are 29 facial boundary curves covering the whole facial area from the chin, mouth to the eyes, eyebrows and forehead. Each facial feature is under control of a group of boundary curves. Each of the boundary curves is represented as a closed loop so that the curves can be formulated using a Fourier series with a period of $0 \leq v \leq 2\pi$. For sake of clarity, only the frontal half of the 29 facial boundary curves are showed discretely in the paper, as the rear half does not contain significant facial information. As shown in Figure 1, we symbolise the boundary curves and PDE surface patches with C and P, respectively. Assume that the two facial boundary curves corresponding to the upper and lower inner lip contours are labeled by C_{22} and C'_{22}. Then the grouping from the top down can be expressed as follows: $P_n(C_{3n+1}, C_{3n+2}, C_{3n+3}, C_{3n+4})$, where $n = (0,1,2,\ldots,6)$, and $P_7(C'_{22}, C_{23}, C_{24}, C_{25})$, $P_8(C_{25}, C_{26}, C_{27}, C_{28})$. As can be observed in Figure 1 (a), there are only 28 isoparametric boundary curves visible. This is because two facial boundary

curves, C_{22} and C'_{22} corresponding to the upper and lower inner lip contours overlap, as highlighted in Figure 1, when the mouth is fully closed. When the mouth opens, these two curves separate, as can be seen in Figure 1 (b).

Like all the parametric models, the resolution and connectivity of the parametric *uv* mesh grid determines those of the reconstructed PDE face. Instead of one *uv* mesh responsible for each of the 9 surface patches, only one *uv* mesh is required for the whole face, making the process more straightforward and intuitive, although the 3D face is generated using nine different PDEs. With the range still remaining $0 \leq u \leq 1$, every one ninth of the whole range of u corresponds to each of the nine surface patches. Taking advantage of parametric representation, the resolution of the PDE face can be flexible. With the same boundary curves it enables a reconstruction of the high-resolution face model by simply increasing the resolution of its *uv* mesh. Generally speaking, the higher the *uv* mesh resolution, the smoother the generated surface. This can be observed in Figure 2, where two PDE faces are produced with the same boundary curves in Figure 1 (a) but with two *uv* mesh grids in different resolutions. This multiresolutional characteristic is one of the advantages of the PDE face over the conventional face models with a fixed LOD in geometry parameterisation.

4.2. Texturing

The PDE face can also be textured on demand. A method, called *uv* mesh texture mapping is developed to help texture the generic PDE face, where correspondence is created directly between the PDE *uv* mesh grid and the texture map, completing the mapping operation within two dimensions throughout, rather than traditional texture mapping between the 3D coordinates and 2D texture map.

Before texturing, we need to make sure that the key features on the texture map align with those on the uv mesh so that correspondence can be created. To do this we warp the texture map using two-pass mesh warping introduced in [37]. The warping process is carried out as a cascade of two orthogonal 1D transformations assisted by one source and one target warp mesh. Figure 3 illustrates the whole warping procedure. Given a texture map as shown in Figure 3 (a), a 7-by-7 source warp mesh, involving 25 most salient facial feature points and 24 sub-feature points on the boundaries of the texture map, can first be manually identified (See in Figure 3 (b)). The 25 facial feature points embrace not only the facial boundary, but the facial features, such as the centers of the eyebrows, eyes and mouth etc; while the 24 sub-feature points are in turn assigned onto the horizontal and vertical extensions of those facial feature points. Figure 3 (c) shows a *uv* mesh scaled to align with the eyes and mouth of the original texture, which is used to assist the

identification of the target warp mesh shown in Figure 3 (d). Once both the source and target warp meshes are determined, two-pass mesh warping can be carried out. Figure 3 (e) shows the finally warped texture map using the target warp mesh. With the warped texture map, the *uv* mesh texture mapping is carried out. Figure 3 (f) shows the textured PDE face.

5. PDEs-driven animation

The conventional direct parameterisation schemes for face model animation, especially those with high-resolution polygon meshes, employ control points to animate the large number of vertices with the assistance of extra interpolation methods. Instead of using the extra interpolation methods, our PDEs-driven facial animation scheme only utilises the boundary curves to animate the high-resolution 3D face. Taking advantage of parametric representation, the PDE method interpolates the animation of the PDE face vertices instinctively. The smooth property of the PDE method guarantees a smooth animation of a high-resolution face model manoeuvred by a few boundary curve points. Figure 4 illustrates the impact of the PDEs-driven animation on the PDE face. The neutral facial boundary curves are discretely shown in Figure 4 (a), with a rectangle highlighting an active area, the enlarged projection view of which on a reconstructed PDE face mesh is shown in Figure 4 (b). The resolution of the PDE face is set to 56-by-40 for the sake of visualisation. Figure 4 (c) shows that, in the rectangle the lower curve is moved towards the upper one with other facial boundary curves remaining fixed. The impact of PDEs-driven animation can be seen in Figure 4 (d), which gives rise to a stretch of a group of vertices. It is worth noting that the magnitude of the vertex stretch is gradually weakening from the bottom up, producing a smooth change in the face surface. This can also be justified by Figure 4 (e), where a motion vector field of such an animation is visualised in its 2D orthogonal projection.

In this section, we propose a direct parameterisation scheme for facial animation by associating PDE boundary curves with a group of animation parameters. In order to make the developed system conform to the latest multimedia standard, we employ the MPEG-4 FAPs so that our system can be used in most contemporary multimedia applications, such as 3D model-based video coding, video conferencing etc. The developed method allows one animation scheme, or one FAT [13] to be used for multiresolutional face models without any extra process.

5.1. FAPs-driven facial animation

In this paper 22 low-level FAPs have been implemented, covering a variety of facial features, such as the nose (FAP 61-64) and cheeks (FAP 39-42, 51, 52) occasionally used in the

exaggerated cartoon expressions, as well as the eyebrows (FAP 31-36), eyes (FAP 19,20) and mouth (FAP 51-54, 59, 60) that contribute most to facial expressions. To implement these FAPs, we first manually identified a group of feature points from the frontal half of the PDE facial boundary curves in accordance with the MPEG-4 FDP feature points. Then, we devise animation rules of how the 22 FAPs affect their corresponding feature points.

To achieve smooth animations, our system not only allows the FAPs to animate their corresponding feature points, but also allows them to affect the neighboring boundary curve points around these feature points. For different FAPs, the animation rules vary according to a study of the facial anatomy. We use a piecewise linear deformation scheme reported in [13] for animation of the mouth, nose, eyes and eyebrows, which provides a flexible simulation of the facial animation. For cheek animation, a linear interpolation is exploited since cheeks are contracted mainly by the zygomatic major, which is a linear muscle [23].

In our method, each FAP is associated with a selected group of PDE facial boundary curve points. There are some PDE boundary curve points which are influenced by more than one FAP. For instance, a group of boundary curve points representing the left eyebrow may be activated simultaneously by FAP31, FAP33 and FAP35. Therefore, a well-designed animation rule should function and make the FAPs interact smoothly when all the three FAPs act on the same boundary curve points, so that there would be no inaccurate animation occurring. We developed an animation scheme to address this problem. Taking the left eyebrow as an example (See Figure 5), this scheme assumes that the rise of two corners of the left eyebrow caused by FAP31 and FAP35, and the rise of the middle eyebrow caused by FAP33 behave like two sinusoids in an inverse and staggered fashion. When only FAP33 functions, the whole curve bends up and down sinusoidally, the trajectory of which is directed by the black dashed arrows in Figure 5. When we want to raise the inner or the outer of the eyebrow, FAP31 or FAP35 will be switched on, respectively, causing the rise of the left eyebrow corners in an inverse way with respect to the movement trajectory triggered by FAP33, as highlighted in grey. This has the advantage that no matter what magnitudes these three FAPs are, the accumulated strength of this group of boundary curve points will not exceed the maximum magnitude of the sinusoid defined. Furthermore, a sinusoid guarantees a smooth simulation of the eyebrow animation.

5.2. Mouth animation of the PDE face

There are four boundary curves involved in mouth animation, as illustrated in Figure 6. The four boundary curves are C_{21}, C_{22}, C'_{22}, C_{23} (the labelling is described in Section 4.1), which are underlined in black in Figure 6. Since Curve C_{22}, and C'_{22} are overlapping when the mouth is

closed, there appear only three dotted lines in the figure. Two groups of boundary curve points on C_{22}, and C'_{22} corresponding to the inner mouth lip contours move off each other when the mouth is open. The motion trajectories are illustrated in Figure 6, using the grey dotted parabolas with the arrows indicating the motion directions. Furthermore, the interactive animation rule for the eyebrows mentioned before is similarly applied to mouth animation of the PDE face. This guarantees that no inaccurate animation will take place when FAP51, 52, 59 and 60 act simultaneously on the mouth.

6. Expression synthesis and transfer

For ease of manipulation, we developed a stand-alone and user-friendly interface with the Microsoft Foundation Classes (MFCs), called the PDE face generator, as shown in Figure 7. This interface allows users to handily synthesise, texture and animate the PDE face for testing purposes. This interface consists of two picture controls on the two sides, giving users an intuitive view of the PDE boundary curve points and an a priori textured PDE face, respectively. Between the two picture controls is a manipulation panel, consisting of two sections. The upper section is a multi-tab control with 22 sliders embedded, each of which specifies one FAP developed according to the MPEG-4 standard, while the lower section embraces six button controls corresponding to six different functions of the PDE face generator. By adjusting the 22 sliders to animate the PDE facial boundary curves can various facial expressions be real-time displayed.

Researchers have concluded six universal categories of facial expressions, i.e. sadness, anger, joy, fear, disgust, and surprise. Any facial expression can be constructed by deforming these six primary expressions with variant intensities [23]. We synthesised the six expressions with our facial geometry parameterisation scheme based on the predefined FAPs. The six expressions are defined with high-level textual descriptions, while the FAPs are a group of low-level parameters, so a method that uses a hierarchical structure tree to associate the high-level expressions with the low-level FAPs is adopted [16]. Table 1 lists the six universal facial expressions with their corresponding textual descriptions and FAPs involved. Figure 8 shows the synthetic results. The first row shows the boundary curve animation governed by the FAPs for the six expressions. The second row shows the reconstructed PDE wireframes in a resolution of 29-by-40. The third row shows the PDE faces using the texture map in Figure 3 (e).

Expression transfer is to use a target face model to simulate a sequence of facial expressions captured and transferred via specific animation parameters from animation of a source character. Expression transfer can be tested by either plugging our facial geometry parameterisation

scheme into any existing MPEG-4 compliant expression capture system, or having the developed system read the MPEG-4 FAP data filed by expression capture systems directly. Here we carry out expression transfer tests by inputting some .fap files available in the test data set of MPEG-4 to the developed system. The MPEG-4 standard defines and provides a number of .fap files, containing a sequence of animation frames in terms of relevant FAP values for test purposes [12]. The PDE face generator can read the .fap files, display consecutively the change of PDE facial boundary curves according to the FAPs read from each frame of the files, and render the PDE face subsequently.

To make the test more vivid, we texture the PDE face with a publicly available face database from the California Institute of Technology (Caltech) [38]. It contains 450 frontal face images of 27 people. Those chosen were manually trimmed into a 256-by-256-pixel resolution with the face centered in each image before texturing.

Figure 9 shows five frames picked when reading the .fap file “marco20.fap”, exhibiting the animation of eye blinking. The first row shows the animating boundary curves of the PDE faces. The second row shows the 2D orthogonal projections of facial animations on the motion vector field, reflecting motion magnitudes and orientations between the neutral and animated PDE face wireframes in a 56-by-40 resolution. The main reason for using the vector field here is that animations can be more easily seen on the vector field than on the dense wireframe. Moreover, the animation smoothness is more intuitively reflected on the vector field than on the wireframe. The third and fourth rows, respectively, show the animation results synthesised with the texture map in Figure 3 (e) and the one obtained from the Caltech database. The FAPs involved here are FAP19, 20, 31-36, 51, 52 53, 54, 59 and 60. Figure 10 shows other examples of five arbitrarily selected frames with texture maps obtained from the Caltech database, when reading the file “expressions.fap”. The FAPs involved here are FAP19, 20, 31-36, 41, 42, 51, 52, 53, 54, 59 and 60.

7. Discussion

In this paper, we propose to parameterise the facial geometry by making use of the PDEs, where the geometry of a generic face is approximated by evaluating spectral solutions to a group of fourth order elliptic PDEs and thus, the face geometry can be represented by a small number of PDE boundary curves in an efficient manner. Compared with those high-resolution face models using direct parameterisation animation methods which need an extra interpolation process [25] [26], the developed method conducts smooth facial animation by using a predefined FAT to associate the MPEG-4 FAPs with the boundary curves, avoiding any extra operation.

Since these boundary curves consist of discrete sample points, no connectivity data is required to represent a face. Generally speaking, the more boundary curve points, the more precise the PDE face, but the more storage space is required. Taking a compromise by choosing 40 points for each PDE facial boundary curve which are sufficient to accurately represent the facial geometry, the whole PDE face requires only 13.9 Kbytes if assigning 3 floats (12 bytes) to each point, while the original face models created by our 3D scanner have an average size of some 1.2Mbytes, which is orders of magnitude larger than the representation of PDE face boundary curves. In general, a 3D laser scanner generated face models may have a size of multimillion bytes. The amount of data required for a PDE face, i.e. the PDE facial boundary curves is much smaller and fixed but can be used to produce a face model in any resolution. Furthermore, as we adopt a spectral approximation to the PDE solution, the storage cost can even be further reduced in a lossy manner, by using only a few coefficients of low frequency modes (Equation 7) which contain the most essential geometric information of the face, instead of using the boundary curves. Thus, a further storage cost reduction without a significant loss of geometric information can be achieved [10], before sending the data for entropy coding.

The intra-patch continuity of the PDE face is secured by its smooth instinct of the PDEs. As for the inter-patch continuity, we employ the four boundary curves as the boundary conditions to the PDE solution for a more precise control, but at the expense of a C^1 inter-patch continuity. In pursuit of the inter-patch continuity, a maximum of tangent plane continuity, which is sufficient in most surface design problems, can be reinforced between two blending PDE patches [30], if the two derivative conditions of each PDE patch are activated. However, as discussed before, this results in a PDE face which does not pass through all the boundary curves, leading to inaccuracy. For a PDE face with a higher continuity, such as the curvature continuity, a sixth order PDE should be sought.

All the tests in this paper are based on the use of a generic PDE face, which has an average geometry of a set of training face data. It is not difficult to synthesise a specific PDE face with the developed parameterisation scheme. To do that, we can either use specific boundary curves acquired from a 3D scanning data set as described previously, or adjust the boundary curves of the generic PDE face to fit a specific human face either manually or automatically. In our work, the PDE face boundary curves are acquired manually from 3D scanning data. Developing an automatic acquisition method is, however, out of focus of this paper as this paper concentrates only on the introduction of a PDE-based facial geometry parameterization scheme. In fact, there exist some similar tactics suitable for such an acquisition process, which can be adopted. For example, the work presented in [39] introduces an automatic PDE boundary curves deriving

scheme for the patchwise PDE method using mesh decimation and calculation of geodesic lines, which ensures the PDE boundary curves acquired from those essential geometric features of the 3D shapes, and makes the PDE approximation close to the original shapes. As for customizing the boundary curves of the generic PDE face, the work in [16] can be employed, which reports an automatic method of how to fit a generic face model to a head-and-shoulder image.

Our algorithm associates the FAPs with some feature points on the boundary curves, making the facial animation independent of the reconstructed PDE face wireframe, such that a multiresolutional 3D face model can be animated by one fixed animation scheme. Nevertheless, the downside of such an alignment is that during animation the PDE face has to be reconstructed whenever there is a relocation of the boundary curves.

References

[1] M. Bloor, M. Wilson, Generating blend surface using partial differential equations, Computer Aided Design 21, (3) 1989, pp. 165-171.

[2] J. Monterde, H. UGAIL, A general fourth order PDE method to generate Bezier surfaces from the boundary, Computer Aided Geometric Design 23, 2006, pp. 208-225.

[3] M. Bloor, M. Wilson, Representing PDE surfaces in terms of B-splines. Computer Aided Design 22 (6), 1990, pp. 324-331.

[4] M. Bloor, M. Wilson, Using partial differential equation to generate free-form surfaces, Computer Aided Design 22 (4), 1990, pp. 202-212.

[5] H. Du, H. QIN, Free-form geometric modeling by integrating parametric and implicit PDEs, IEEE Trans. on Visualization and Computer Graphics 13 (3), 2007, pp. 549-561.

[6] M. Bloor, M. Wilson, The efficient parameterisation of generic aircraft geometry, Journal of Aircraft 32 (6) 1995, pp. 1269-1275.

[7] C. Dekenski, M. Bloor, M. Wilson, The computer-aided functional design of a marine propeller, Journal of Ship Research 40 (2), 1996, pp. 117-124.

[8] G. Castro, H. Ugail, Shape morphing of complex geometries using partial differential equations, Journal of Multimedia 2 (6), 2007, pp.15-25.

[9] H. Ugail, A. Sourin, Partial differential equations for function based geometry modelling within visual cyberworlds, Cyberworlds (2008), IEEE Computer Society, pp. 224-231.

[10] Y. Sheng, A. Sourin, G. Castro, H. Ugail, A PDE method for patchwise approximation of large polygon meshes, The Visual Computer vol. 26, no. 6-8, 2010, pp. 975-984.

[11] P. Eisert, B. Girod, Analyzing facial expression for virtual conferencing, IEEE Computer Graphics and Application 18 (5), 1998, pp. 70-78.

[12] ISO/IEC IS 14496-2, MPEG-4 Visual. 1999.

[13] A. Tekalp, J. Ostermann, Face and 2D animation in MPEG-4, Signal Processing: Image Communication 15, 2000, pp. 387-421.

[14] F. Parke, A Parametric Model for Human Faces, Tech. Report UTEC-CSc-75-047, University of Utah, 1974.

[15] J. Ahlberg, CANDIDE-3: An Undated Parameterised Face, Report No. LiTH-ISY-R-2326, Linkoping University, Sweden, January 2001.

[16] Y. Sheng, A. Sadka, A. Kondoz, Automatic single view-based 3D face synthesis for unsupervised multimedia applications, IEEE Trans. on CSVT 18 (7), 2008, pp. 961-974.

[17] G. Abrantes, F. Pereira, MPEG-4 facial animation technology, survey, implementation, and results, IEEE Trans. CSVT 9 (2), 1999, pp. 290-305.

[18] M. Patel, P. Willis, The facial animation, construction and editing system, Proc. of Eurographics'91 (1991), pp. 33-45.

[19] K. Waters, A muscle model for animating three-dimensional facial expression, Computer Graphics 21 (4), 1987, pp. 17-24.

[20] D. Terzopoulos, K. Waters, Analysis and synthesis of facial image sequences using physical and anatomical models, IEEE Trans. on PAMI 15 (6), 1993, pp. 569-579.

[21] P. Ekman, W. Friesen, Facial Action Coding System, Consulting Psychologist Press, 1977.

[22] M. Rydfalk, CANDIDE, a Parameterised Face, Report No. LiTH-ISY-I-866, University of Linkoping, Sweden 1987.

[23] F. Parke, K. Waters, Computer Facial Animation, Wellesley, Massachusetts, 1996.

[24] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, In Proc. SIGGRAPH'99 (1999), pp. 187-194.

[25] W. Lee, N. Magnenat-Thalmann, Fast head modeling for animation, Image and Vision Computing 18, 2000, pp. 355-364.

[26] B. Yin, C. Wang, Q. Shi, Y. Sun, MPEG-4 compatible 3D facial animation based on morphable model, Proc. of the 4th Int. Conf. on Machine Learning and Cybernetics, 2005, pp. 4936-4941.

[27] D. Luebke, A developer's survey of polygonal simplification algorithms, IEEE Computer Graphics and Applications 21 (3), 2001, pp. 24-35.

[28] N. Ersotelos, F. Dong, Building highly realistic facial modeling and animation: A survey, The Visual Computer, vol. 24, 2008, pp. 13-30.

[29] Z. Deng, J. Noh, Computer facial animation: A survey, Data-driven 3D facial animation, Springer, 2007, pp. 1-28.

[30] H. Ugail, M. Bloor, M. Wilson, Techniques for interactive design using the PDE method, ACM Transactions on Graphics 18 (2), 1999, pp.195-212.

[31] C. Dekenski, M. Bloor, M. Wilson, Partial differential equation surface generation and functional shape optimisation of a swirl port, AIAA Journal of Propulsion and Power 13, 1997, pp. 398-403.

[32] L. You, J. Zhang, P. Comninos, Generating blending surfaces with a pseudo-Levy series solution to fourth order partial differential equations, Computing 71 (4), 2003, pp. 353-373.

[33] J. Zhang, L. You, Fast surfaces modelling using a 6th order PDE, Computer Graphics Forum 23 (3), 2004, pp. 311-320.

[34] S. Kubiesa, H. Ugail, M. Wilson, Interactive design using higher order PDEs, The Visual Computer 20 (10), 2004, pp. 682-693.

[35] H. Du, H. Qin, Direct manipulation and interactive sculpting of PDE surfaces, Computer Graphics Forum 9 (3), 2000, pp. 261-270.

[36] M. Bloor, M. Wilson, Spectral approximation to PDE surfaces, Computer-Aided Design 28 (2), 1996, pp. 145-152.

[37] G. Wolberg, Digital Image Warping, IEEE Computer Society Press, Los Alamitos, CA, 1990.

[38] http://www.vision.caltech.edu/html-files/archive.html

[39] M. Pang, Y. Sheng, A. Sourin, G, González Castro, H. Ugail, Automatic reconstruction and web visualization of complex PDE shapes, In Proc. of 2010 Int. Conf. on Cyberworlds, Singapore, 20-22 Oct, 2001, pp. 97-104.

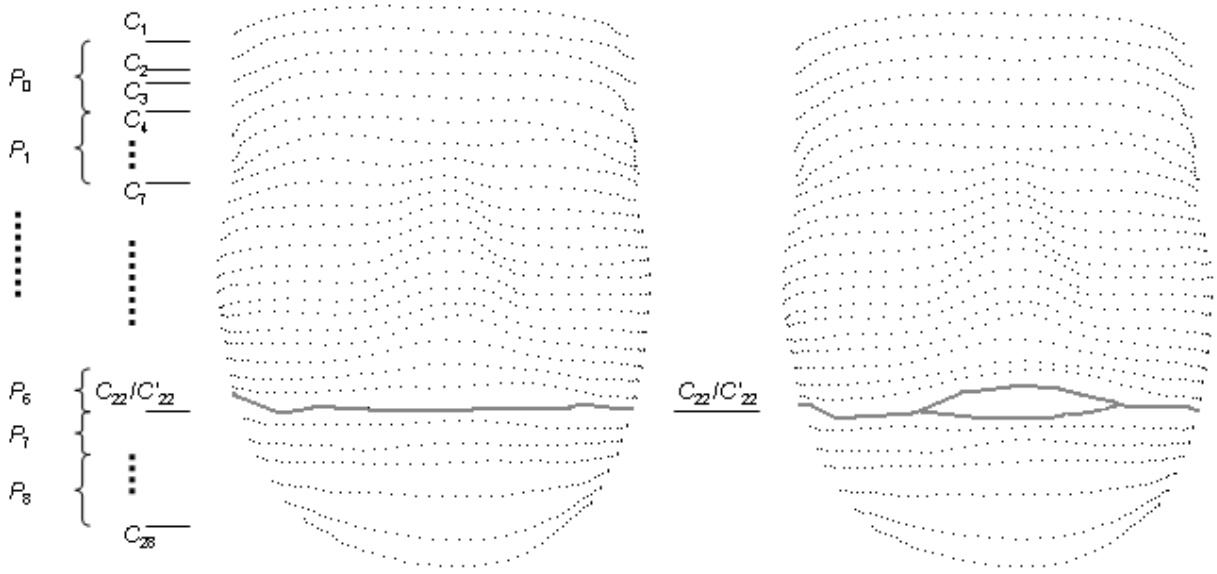
Table and Figures

Table 1: Textual descriptions of six universal facial expressions with the FAPs involved.

TABLE I

Expression Name	Textual Description	FAPs
Joy	The eyebrows are relaxed. The mouth is open and the mouth corners pulled back toward the ears.	51-56
Sadness	The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is relaxed.	19,20,31,32,35, 36
Anger	The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other.	19,20,31,32,35, 36
Fear	The eyebrows are raised and pulled together. The inner eyebrows are bent upward. The eyes are tense and alert.	19,20,31-36,51,52,59,60
Disgust	The eyebrows and eyelids are relaxed. The upper lip is raised and curled, often asymmetrically.	51,52,59,60
Surprise	The eyebrows are raised. The upper eyelids are wide open, the lower relaxed.	19,20,31-36,51-54,59,60

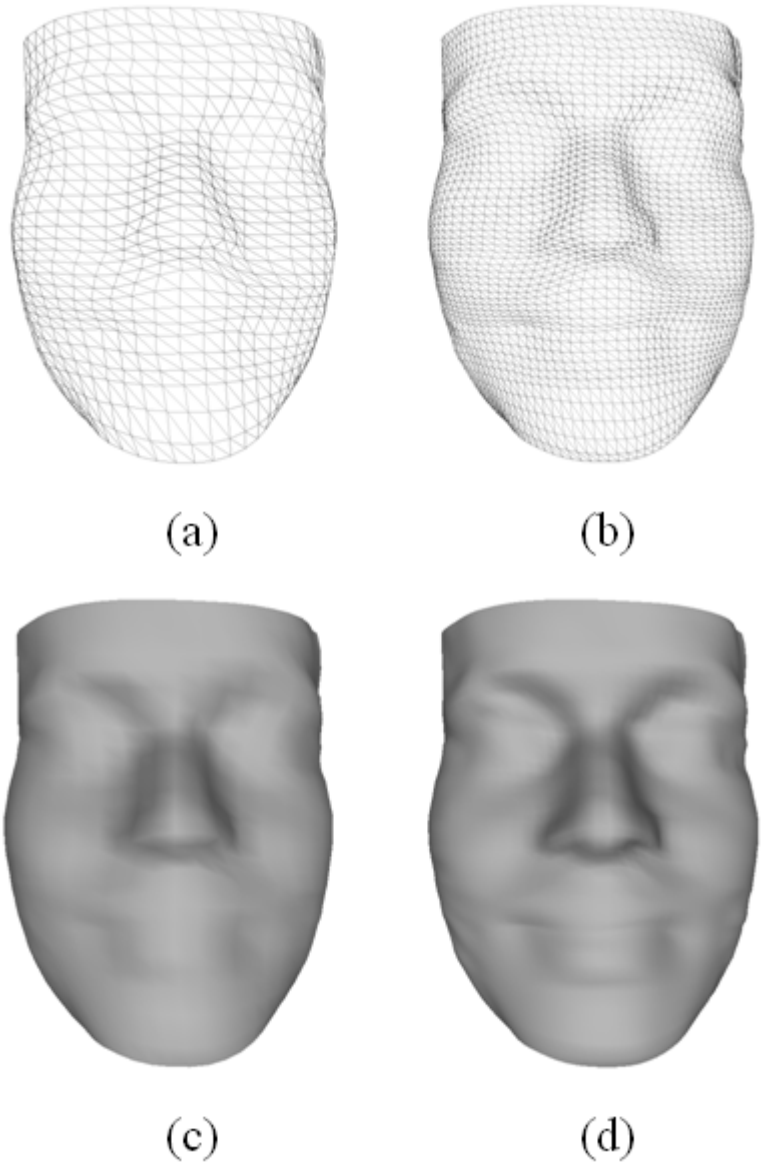
Figure 1: 29 boundary curves of the PDE face.



(a) Labelled boundary curves with the mouth closed

(b) boundary curves with the mouth open.

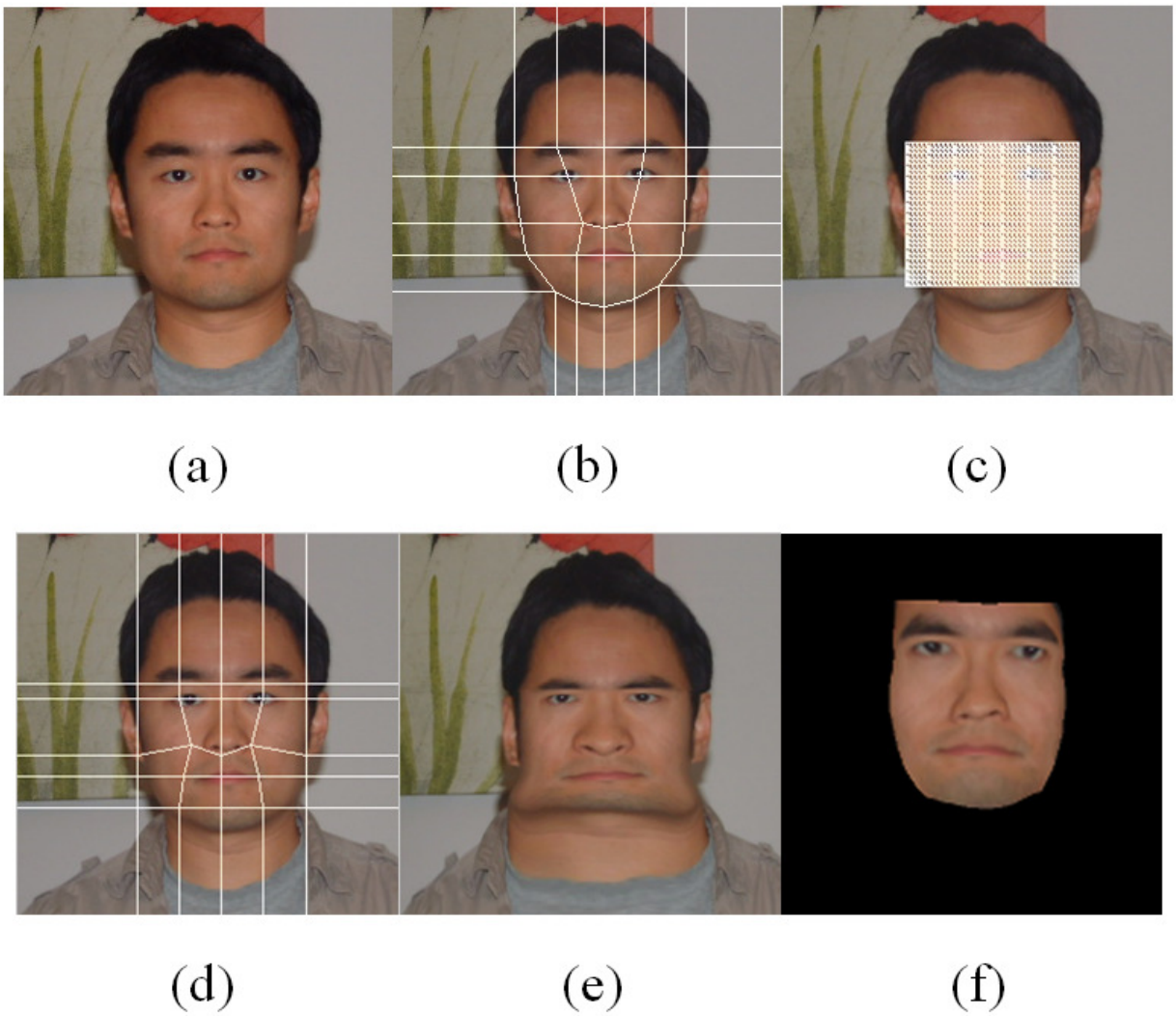
Figure 2: The PDE face produced in different resolutions with the same boundary curves.



(a) and (b) are wireframes in uv resolutions of 28-by-20 and 50-by-40, respectively.

(c) and (d) are the corresponding shaded faces.

Figure 3: Illustration of the texture map warping process.



(a) Original texture,

(b) original texture with source warp mesh,

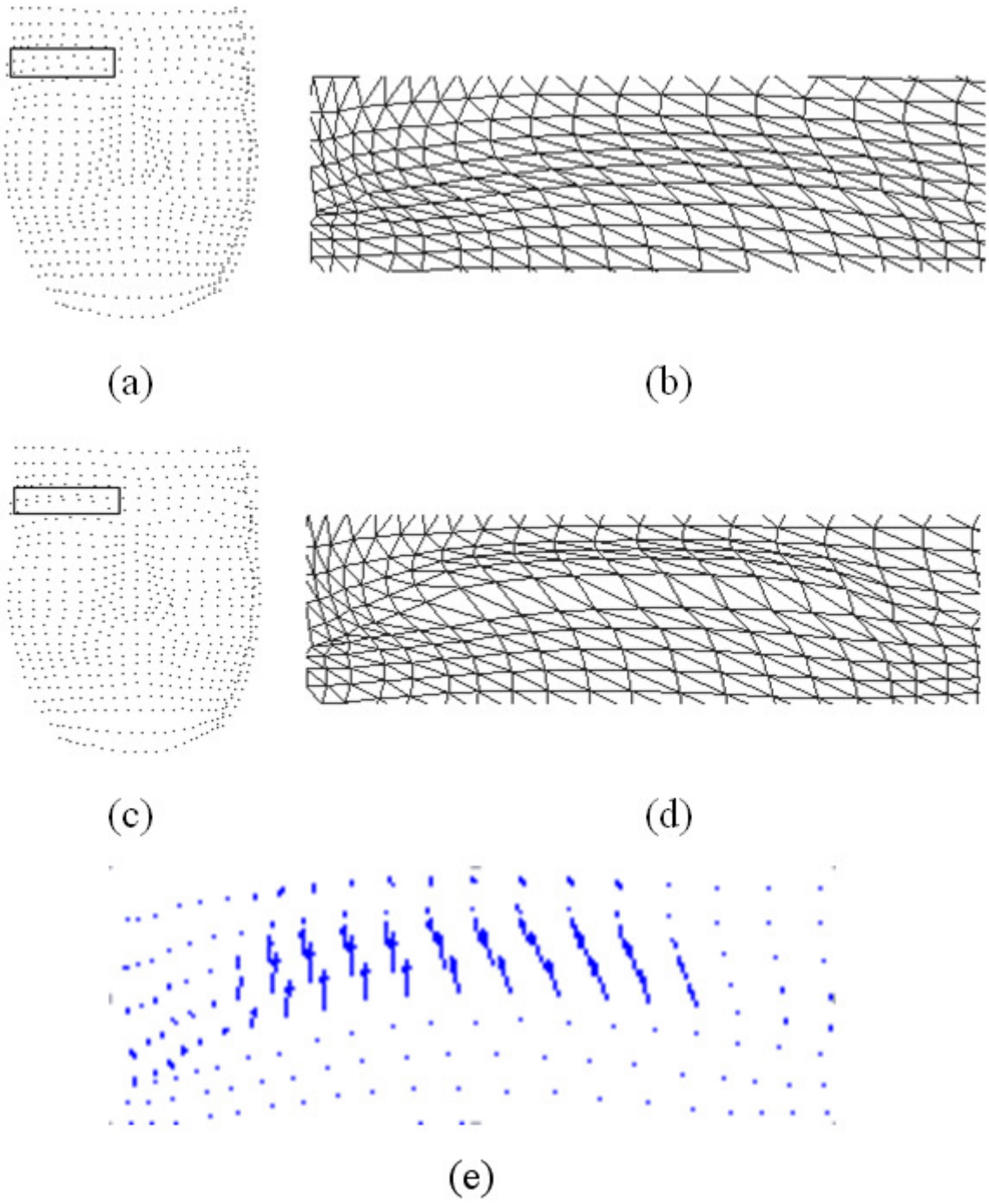
(c) original texture with scaled uv mesh,

(d) original texture with target warp mesh,

(e) warped texture,

(f) textured PDE face.

Figure 4: PDEs-driven animation.



(a) Facial boundary curves before animation,

(b) active area on the PDE face before animation,

(c) facial boundary curves after animation,

(b) active area on the PDE face after animation,

(e) motion vector field of the active area.

Figure 5: Illustration of the left eyebrow animation in a sinusoidal fashion.

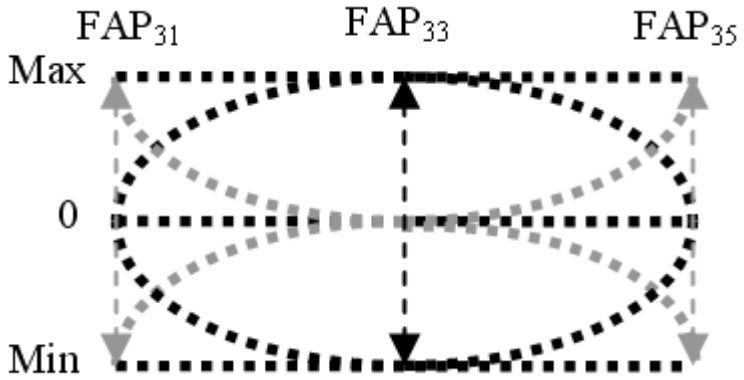


Figure 6: Facial boundary curves for mouth animation.

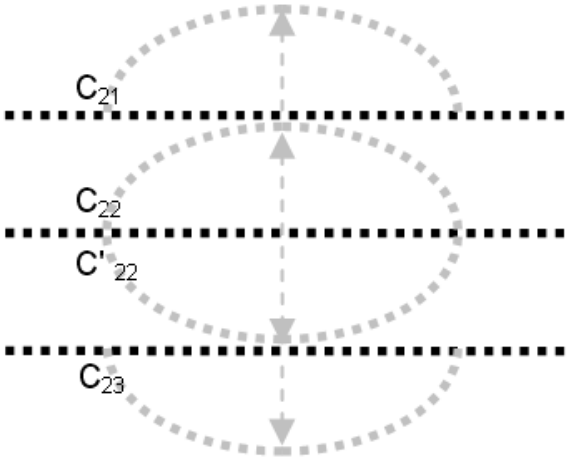


Figure 7: PDE face generator.

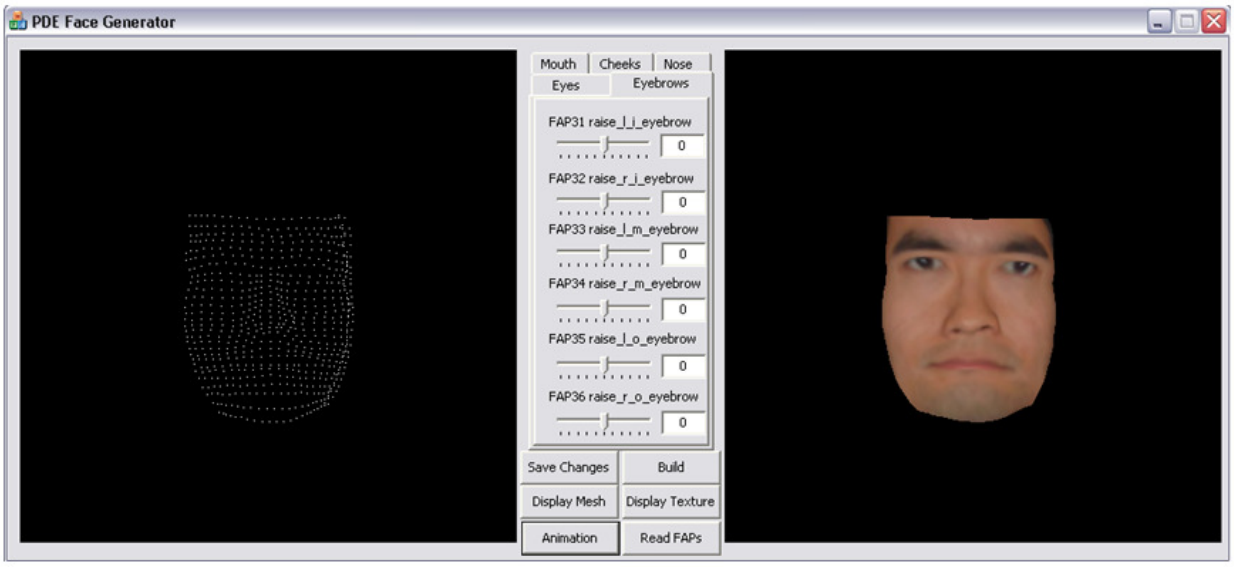


Figure 8: Implementation of six universal facial expressions with the PDE face in different resolutions: Joy, sadness, anger, fear, disgust, surprise.

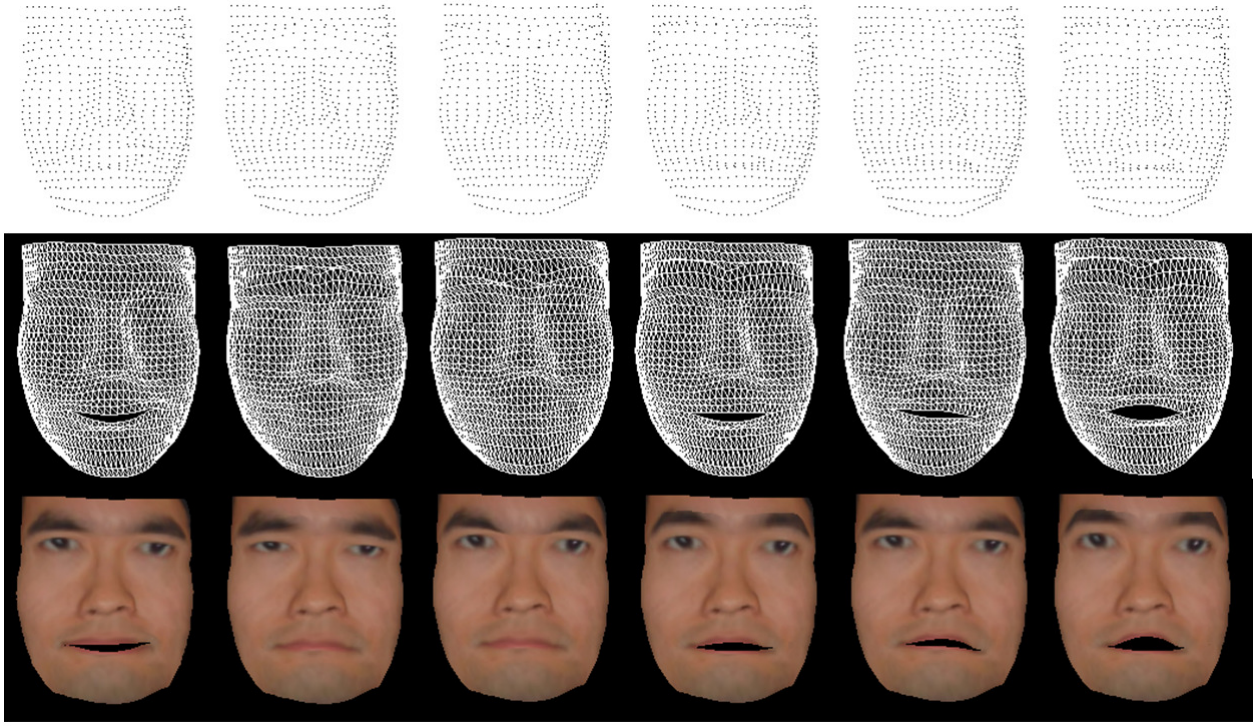


Figure 9: Expression transfer by reading frame 100, 120, 125, 130 and 150 of the FAP file "marco20.fap".

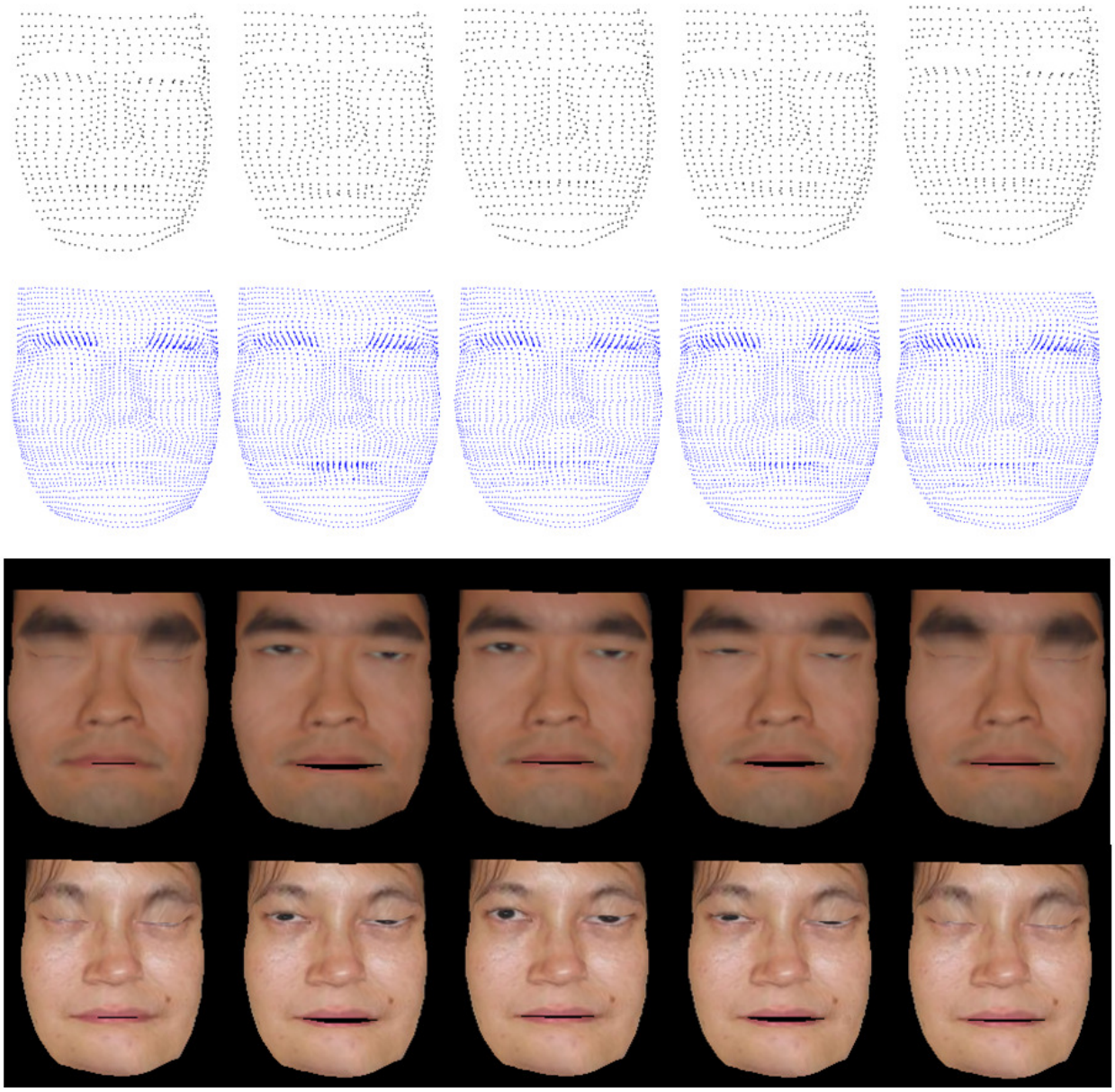


Figure 10: Expression transfer by reading frame 30, 50, 100, 200 and 300 of the FAP file "expressions.fap".